REMARKS

Claims 5, 6, 10-12 and 14-21 are pending in the instant application (the "'320 Application"). Claims 5, 10-12 and 14 are amended herewith. Claims 1-4, 7-9 and 13 are cancelled without prejudice, and may be pursued in a continuing application. As detailed below, it is believed that the above amendments and the following remarks address and resolve each rejection presented in the Office Action mailed 24 December 2009.

Claim Amendments

Claims 5 and 14 are amended in accordance with the Examiner's indication of allowable subject matter. In particular, claim 5 is amended to incorporate all features of claim 7. This is equivalent to amending claim 7 to rewrite the claim in independent form, as required by the Examiner for allowance. Claim 14 is rewritten in independent form, to include all features of base claim 5. Per the Notice of Allowable Subject Matter (see page 3, items 9-11 and page 8, item 17 of the pending Office Action) and the arguments in response to the §112 rejection, below, these amendments place the claims in condition for allowance.

Given the cancellation of claim 8, claims 10-12 are amended to depend from claim 5. No new matter is added.

Accordingly, claims 5, 10-12, 14 and 15 are all believed allowable. Claim 6 is also believed allowable at least because it depends from amended claim 5.

New Claims

New claims 16-18 and 19-21 are reiterations of claims 10-12, with claims 16-18 depending from claim 14 and claims 19-21 depending from claim 15. Claims 16-18 do not add any new matter. Claims 16-21 are believed allowable because they depend from allowable base claims (per the Examiner's indication of Allowable Subject Matter and the amendments made above).

Claim Rejections – 35 U.S.C. § 112

All pending claims stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite, in particular because the term "large prime numbers" is considered by the Office to be a relative term.

Applicant respectfully disagrees. A "large prime number" is and has been understood in the art of computer science as having a specific meaning. For example, currently, a prime is "large" if it has 2048 or more bits. See P. Lutus, "Prime Numbers, Exploring a unique class of numbers" at page 4, paragraph 11 (boxed in red), included as Appendix A. It should be noted that the size of a "large" prime changes with the performance of computers, as is also known in the art.

Applicant submits that the meaning of "large prime number" would be understood by one of skill in the art. Withdrawal of the §112 rejection is respectfully requested.

Claim Rejections – 35 U.S.C. § 102 - Vallee

Claims 5 and 6 stand rejected under 35 U.S.C. §102(e) as being anticipated by Vallee. Applicant continues to disagree and traverses the rejection for reasons of record.

However, per the Examiner's notice of allowable subject matter, the amendment to claim 5 (incorporating all features of claim 7) overcomes the rejection of claim 5. Since claim 6 depends from claim 5, it is also not anticipated by Vallee. Applicant therefore respectfully requests withdrawal of the Section 102 rejection of claims 5 and 6.

Claim Rejections – 35 U.S.C. § 103 – Bartram in view of "Admission"

Claims 1, 3, 8 and 13 stand rejected under 35 U.S.C. Section 103(a) as being unpatentable over U.S. Patent Publication No. 2004/0054885 (hereinafter, "Bartram"), in view of a purported "Admission" found at pages 1-3 of the '320 Application.

<u>Claims 1, 3, 8 and 13 are cancelled herewith; therefore, this §103 rejection is fully addressed</u>.

However, for the record, Applicant continues to disagree with the rejection for the following reasons, further detailed in the Response of 21 September 2009:

1. The "Summary of the Invention" section is improperly asserted as prior art;

2. The cited art itself does not support the assertion of obviousness, and no extrinsic evidence is provided to support the assertion of obviousness, and
3. Bartram teaches away from use of zero knowledge authentication

Claim Rejections – 35 U.S.C. § 103 – Bartram in view of "Admission" and Vallee

Given the rejection of claims 2, 4 and 9, claims 10-12 stand rejected under 35 U.S.C. Section 103(a) as being unpatentable over Bartram and the purported Admission described above, and further in view of Vallee.

<u>The rejection is overcome by the cancellation of claims 2, 4 and 9 and the amendments to claims 10-12</u> (these claims now depend from claim 5, amended per the Examiner's indication of allowable subject matter). Claims 10-12 are allowable at least because they depend from an allowable base claim, and Applicant therefore respectfully requests withdrawal of this §103 rejection.

The cancellation of claims 2, 4 and 9 and the amendments to claims 10-12 do not indicate agreement with the pending rejection. For the record, and as further detailed in the Response of 21 September 2009, Applicant reiterates:

4. The "Summary of the Invention" section is improperly asserted as prior art;
5. The cited art itself does not support the assertion of obviousness, and no extrinsic *evidence* is provided to support the assertion of obviousness, and
6. Bartram teaches away from use of zero knowledge authentication
7. Adding Vallee to the combination used to reject claim 1 does not solidify the rejection.
8. The cited Vallee passage does not disclose first and second authentication agents or generating a *new* secret, as required in claim 2.
9. Vallee and the additional cited art does not generate a secret *to have a value relatively prime to the product*, as in claim 4.
10. The cited art does not teach generating a new secret, as required in claim 9.
11. The cited art does not teach installing authentication and prover agents on each computer through common software, as in claim 12.

CONCLUSION

The above amendments and remarks address and overcome each rejection presented in the office action of 24 December 2009, placing the '320 Application in condition for allowance. Applicants thus respectfully solicit a Notice of Allowance for all pending claims.

No fees are due for new claims 16-21, since the '320 Application now includes a total of 13 claims (3 independent and 10 dependent), which is within the number of claims included in the filing fee paid by Applicant. No other fees are believed due; however, if any additional fee is deemed necessary in connection with this paper, please charge the aforementioned Deposit Account. Should any issues remain outstanding, the Examiner is encouraged to telephone the undersigned agent.

Respectfully submitted,
LATHROP & GAGE LLP

Date: 21 March 2010

By: Heather Perrin

Heather Perrin, Reg. No. 52,884
4845 Pearl East Circle, Suite 201
Boulder, Colorado 80301
Tel No: (720) 931-3033
Fax No: (720) 931-3001

APPENDIX A

Response to non-final Office Action of 12/24/2009

U.S. Serial No. 10/687,320

Atty. Docket No. 413130

Prime Numbers

Exploring a unique class of numbers

— P. Lutus — Message Page —

(double-click any word to see its definition)

Introduction

This may be hard to believe, but there's a special class of numbers that influence many things in the modern world, including cryptography and the behavior of locusts. As to the first, a popular encryption scheme uses prime numbers to create a very good level of security (but one that may erode in the future because of a new kind of computer). As to the second, locusts aren't mathematicians, but nature makes them pay attention to prime numbers anyway.

Prime numbers are no less than the foundation on which ordinary counting numbers (0,1,2,3, ...) are built. As it turns out, each positive integer larger than 1 is either itself prime, or is composed of a unique set of prime factors (this is called the fundamental theorem of arithmetic). Numbers composed of prime factors are called "composites". For example:

- 99981599 is prime.
- 99981600 is composite, equal to $2^5\ 3\ 5^2\ 41659$ (notice about this example that 2, 3, 5, and 41659 are all prime numbers).
- 99981601 is prime.

Here is an online calculator — enter a number and press "Factor" to compute its prime factors:

JavaScript Prime Factor Calculator

Enter a number: 99981600 Factor

Result: composite: $2^5\ 3\ 5^2\ 41659$ (< 0.001 seconds)

Calculator Notes:

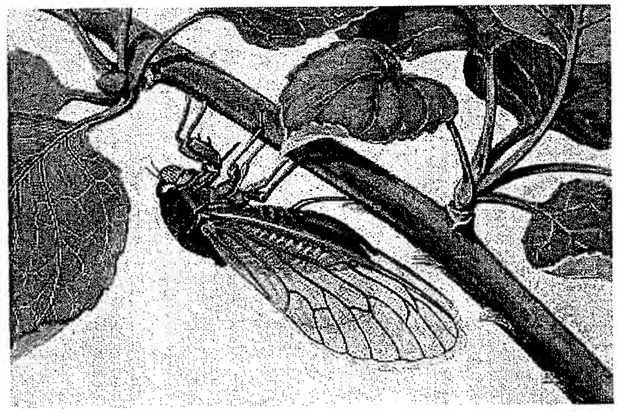
> Notice about the calculator's output that two adjacent numbers imply multiplication — the list of factors for the default example is "$2^5\ 3\ 5^2\ 41659$," which if fully spelled out would be "2 × 2 × 2 × 2 × 2 × 3 × 5 × 5 × 41659". The use of exponential notation ($2^2 = 2 \times 2$) is just a way to shorten the display of repetitive factors. To see the point of this, enter 4503599627370496 as a number to factor above. The result (2^{52}) means "two multiplied by itself 51 times."
>
> Factoring takes the least amount of time for composite numbers, and primes require the most computation time (remember this for a later discussion). If you type in a difficult number (a number that is prime and/or has a lot of digits) and if your computer is not fast, after a few seconds most browsers will complain and ask whether you want to stop the calculation.
>
> The worst-case prime for this calculator is 9007199254740881, the largest prime below 2^{53}-1

(the upper numerical limit for this calculator). On some browsers (but not all) this value will produce a long computation and may cause your browser to complain — but the calculator will eventually identify it as prime. Because JavaScript is a relatively slow interpreted language, on a modern desktop and depending on the selected browser, this particular computation might require as much as 45 seconds (a C version of this program requires only 1/3 second).

The increasing number of applications for prime numbers shows that, as our understanding of the world increases, we accept more and more mathematical ideas, even from fields like number theory that had until recently been regarded as having no practical significance. In this article we will explore prime numbers and see how they play an increasingly important part in modern life and in the workings of nature.

Mathematical Locusts

I will have more to say about primes from a technical standpoint, but I first want to show that prime numbers aren't just an esoteric area of study with no connection to reality. As we study nature, we discover more and more examples where mathematics and algorithms occupy important chairs in nature's orchestra. Here is an example.



Magicicada septendecim (R. E. Snodgrass)

It is well-known that a certain kind of locust (actually a cicada) spends most of its time in hiding, only reappearing to mate every 13 or 17 years (13 years in the southern U.S., 17 years in the north). This reproductive pattern has been noted by many, but until recently no one understood why. Obviously by hiding so long, these cicadas sacrifice opportunities to eat and breed, which leads one to ask — what do they get in return?

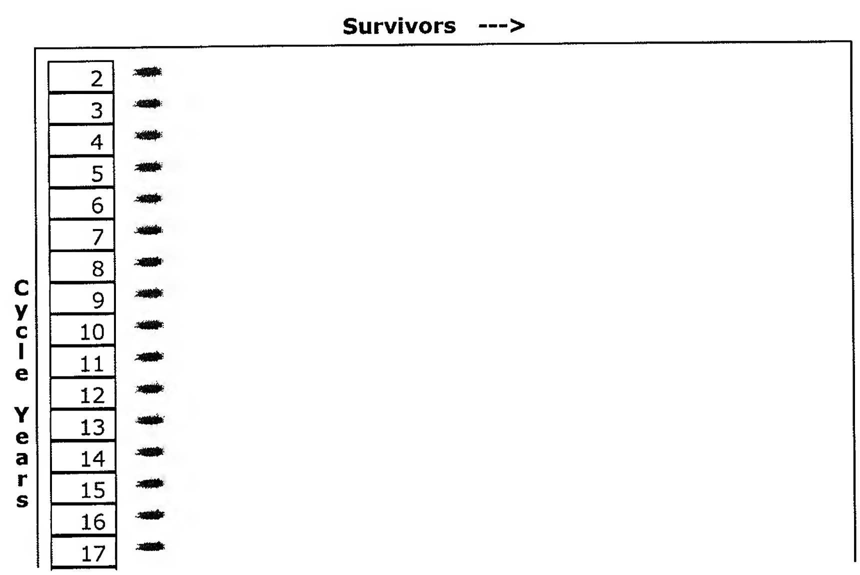
It's important to remember that among living species, a behavior is ultimately judged by whether it confers a survival advantage — if it doesn't, that species will either disappear or evolve away from the behavior. So the persistence of the 13- and 17-year cicada reproductive cycles must confer a survival advantage. What is it?

Let's look at this from the standpoint of a predator. To a predator, it would be nice if the cicadas appeared on a regular basis, preferably one synchronized with the predator's own reproductive cycle. And if the cicadas won't coöperate by having exactly the same reproductive cycle, a second choice would be a multiple — if the predator reproduces at, say, four year intervals, the cicadas could play along by reproducing at 4 or 8 or 12 or 16 year intervals — any of those would be coöperative and helpful, because some of the cicadas would show up just when the predator does.

But from the perspective of the cicada, the best reproductive cycle would be one that *minimizes* possible interactions with predators — a cycle that would confuse and thwart predators. The best choice would be a reproductive cycle that minimizes interactions with *any* predators, regardless of their own reproductive cycles.

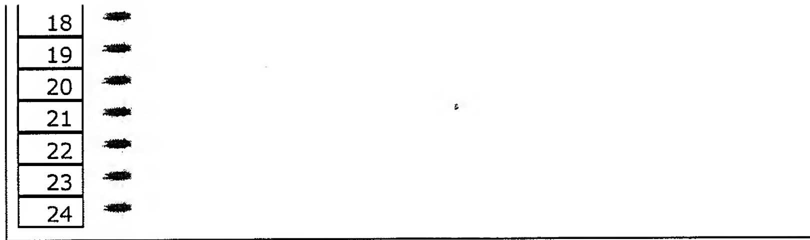
That's asking a lot. Multiple predators have different reproductive cycles, and worse, over time predators may evolve and adapt to any cycle the cicadas choose. It turns out that *any predator reproductive cycle that evenly divides the cicada's cycle* means the predator will hatch out in synchrony with the cicada at least some of the time. And some numbers are worse than others. For example, if the cicadas evolved toward a 12-year reproductive cycle, this would work to the advantage of predators that reproduced at 2,3,4, and 6 year intervals, so 12 would seem to be a particularly bad choice.

Given a large set of predators, any of which might drift toward matching the cicada's reproductive cycle, what would be a good choice for the cicadas? Well, we know that a prime number has no divisors apart from itself and 1, so a prime number of years would have the effect of *minimizing contact with predators*, regardless of what years the predators appear. This makes 13-year and 17-year reproductive cycles seem like reasonable choices (both 13 and 17 are prime numbers).



Next issue. Can a strategy like this arise by chance, or does it require an intelligent designer ? According to the Theory of Evolution , species evolve toward optimal survival strategies *by chance, not by design*.

So for this prime-number theory to have any credibility, we should be able to build a dumb model that (like evolution) depends on chance, not design, and see which survival strategies *naturally evolve*. The model can't know anything about prime numbers and it can't have any preconceived notions built in.



Cicada Evolution Model: Start Reset **Year 0000**

If such a model were to fail, advocates of intelligent design might argue this supports their cause (they might argue that God talks to cicadas but not to computers). If the model succeeds, this doesn't disprove the idea of an intelligent designer (nothing can), but it shows that a natural (as opposed to supernatural) explanation is available.

By now readers will have noticed that a cicada behavioral model is built into this page. Please press "Start" on the model and watch it run. The simulation models 1000 years of cicada evolution, and it clearly shows a *natural advantage* for reproductive cycles based on prime numbers. The model works like this:

- For both cicadas and predators, the model treats all reproductive cycle times the same. Any differences between years come from the impartial workings of mathematics.
- During a 1000-year simulation (that only takes seconds of computer time), the model correlates cicada survival with reproductive cycle times.
- Each cicada cycle time is compared to all different predator reproductive cycle times, and for each combination, a test is performed to discover whether the predators and cicadas are synchronized.
- The simulation has this rule: when predators and cicadas are present at the same time, *predators eat cicadas*.
- Just as in nature, long cicada reproductive cycle times get a penalty.
- The simulation doesn't know anything about prime numbers.

This simulation is simpleminded in the same way that evolution is simpleminded — predators eat cicadas whenever they can, and natural selection decides who survives. At the end of the simulation, we see a very clear advantage for reproductive cycle times based on primes — all the distinctive peaks in the chart (5,7,11,13,17,19, and 23) are prime numbers.

Notice about the model that it confirms the problem with 12 as a cycle time. Because 12 has a lot of divisors, it would be a bad choice — it readily synchronizes with predators having many different reproductive cycles. The simulation confirms this by giving 12 a low score. It also shows the number 18 has the same problem, and for the same reason — too many divisors (2,3,6, and 9), therefore too many opportunities for predators.

Isn't it interesting that a simple program that basically throws dice can come up with results that imitate nature? In fact, this is just one case of a gradual awakening to the fact that much of nature runs on mathematical and algorithmic principles.

Finding Primes

Making a list of prime numbers is surprisingly easy. One of the simplest methods was created by Eratosthenes , a classical Greek mathematician and what we now would call a polymath . Among many accomplishments, Eratosthenes invented what we now call the "Sieve of Eratosthenes ", a way to quickly distinguish prime numbers from composites.

The Sieve works like this:

1. Create a list of sequential integers, **{1 ... n}**, **n** being the largest number of interest.
2. Initially set a position **p** at the first prime number in the list (the number 2).
3. Strike off all numbers that are multiples of **p**: **{p*2, p*3, ...}**.
4. Move **p** forward in the list to the the nearest number not stricken off in step (3) (this will be a prime number).
5. Until **p** is greater than or equal to the square root of **n**, repeat from step (3) above.
6. Done.

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

Width: 16 Height: 16 Start Reset

The Sieve is trivial to implement in computer code and is useful for creating lists of primes within reasonable values of **n**. It's important to note that the Sieve is normally represented in a computer as an array of binary bits, consequently increasing the value of **n** requires more computer memory. Finally the point is reached where the bit array is too large to be practical and other methods must be used.

I want to emphasize that the Sieve example on this page is represented as a square array only for display convenience. The Sieve is in some ways better visualized as a long one-dimensional row of numbers. As it turns out, because of how I've designed the example, the reader can enter particular width and height numbers to form a long row (or column) and see how this looks. Unfortunately, most of us don't have really, really large computer displays, so most of a long row or column will be invisible at any particular time.

Prime Secrets & Quantum Computing

In this anecdote, clever thinking and cheap computer power have made possible a new application for number theory, an otherwise esoteric field of mathematics. Because of developments in number theory and computer algorithms, sensitive financial transactions and diplomatic messages are now securely transmitted by way of a method that uses prime numbers.

Prior to modern times, message encryption and decryption depended on a private-key system. In the private-key system, a deciphering key is delivered to any intended recipients of secret messages, after which enciphered messages can be transmitted in the open with little risk that they will be decoded by third parties. The Achilles' heel of this system lies in the delivery and maintenance of the private key — if anything goes wrong, if the key is intercepted without the knowledge of the sender or receiver, the system is compromised.

In 1997 a new scheme called "public-key cryptography" was published by Ron Rivest , Adi Shamir , and Leonard Adleman at MIT . Now popularly known as RSA Encryption , it has greatly simplified secure transactions of all kinds. A similar scheme was envisioned in 1973 by by Clifford Cocks , but at the time sufficient computer power wasn't available to make the scheme practical, and the original work was classified, therefore unpublished.

The public-key cryptography system uses two keys (in the form of large numbers), one public and one private. The public key can be revealed without compromising the security of the system, and parties wanting to send a message can encrypt using the public key, making the message secure before transmission. Notice that this system eliminates the requirement to share a private key.

Here is an example of an RSA transaction:

- Bob sets up an RSA system with a public and private key. He makes the public key available to anyone who wants it.
- Alice uses the public key to encrypt a message for Bob. Once encrypted with the public key, the message is quite secure and can be transmitted through normal communication channels.
- Bob receives Alice's message and decrypts it using his private key.

Anyone can use Bob's public key to encrypt a message, but let's say Alice wants Bob to be certain that the message came from Alice. To accomplish this, Alice would encrypt the message with her *private* key, before encrypting with Bob's public key and transmitting. When Bob receives the message, he adds the step of decrypting with Alice's *public* key, thereby confirming that it originated with Alice. Notice in this signed-message variation that Alice doesn't need to reveal her private key.

I hope the reader has noticed that the sender and receiver don't ever have to meet for the system to work, and there is no longer a requirement for secret communication pathways.

The security of RSA is primarily tied to the difficulty of factoring large composite numbers. The first step in configuring an RSA system is to choose two large prime numbers. The next step is to multiply the primes together to create a composite number whose factors are the two original primes.

As it turns out, generating two primes of adequate size is not difficult, nor is multiplying them together. The problem lies in factoring the resulting composite number — given two sufficiently large prime numbers, it is believed to be infeasible to factor the product into its components in a finite amount of time. But over time, because of a gradual increase in computer power, the definition of "sufficiently large" has changed, and the RSA system adapts by increasing the recommended number of binary bits in the prime numbers. At the time of writing, the two primes are regarded as sufficiently secure if they are 2048 bits long. The product of two such primes is twice as long — 4096 bits, or about 1234 decimal digits (see side note).

Math side note

To determine the number of digits in base B (nd_B) for a given number of digits in base A (nd_A), use this equation:

$$nd_B = \frac{nd_A \log(A)}{\log(B)}$$

The reader may remember about the calculator at the top of this page that primes require much more computation time than composites. It turns out that factoring a composite number made up of two large primes is almost as time-consuming. At some risk of oversimplification, the RSA system publicly reveals only the composite number (called an "RSA Number"), not its two components, and this is the key to RSA's security.

Much of modern banking, commerce and diplomacy depends on the security of the RSA system or an equivalent. Confidence in these systems hinges on the assumption that large composite numbers cannot be factored in any reasonable time. Unfortunately, this is an assumption, it hasn't been proven, and over time we see improvements in the efficiency of factoring algorithms.

Those responsible for computer security sometimes organize contests in which people are encouraged to think of new, more ingenious ways to factor composites. The RSA Factoring Challenge is one such contest, in which monetary rewards are offered for the successful factorization of "RSA Numbers" , the keys to the RSA system.

In 2005, a 640-bit RSA number eponymously named RSA-640 was successfully factored by a German team, an effort that required five months and 80 processors. From this and similar results we can conclude that our trust in 4096-bit RSA numbers is well-placed ... for now.

Quantum Computers

If it becomes practical, a new kind of computer may erode this confidence. Some preliminary examples of a "quantum computer" have been tested in laboratories and, although there is still uncertainly that such a scheme can be made reliable, if it succeeds it bodes ill for RSA.

A conventional computer attacks a problem in a series of steps. For example a simple integer factoring program might use this approach to factoring a test number:

1. Start with the first prime number (2).
2. Divide the test number by the prime. If the division has no remainder (meaning the divisor is a factor of the test number), replace the test number with the result and add the divisor to the list of factors.
3. Repeat step 2 until it fails.
4. Choose the next prime number.
5. Test whether the new prime is greater than or equal to the test number's square root. If yes, move to step 6, if not, loop back to step 2 above.
6. Return the list of factors. If the only factor is the test number, the number is prime.

Sophisticated factoring programs use various strategies to improve on this simple algorithm, but basically, as far as is known at the time of writing, one is reduced to laboriously dividing a test number by many, many possible factors, and as the test number grows larger, the problem's severity increases beyond polynomial time . In everyday terms, in a world of conventional computers, schemes like RSA are quite secure.

But quantum computers are not conventional. A quantum computer, if it turns out the be practical, would solve the integer factorization problem by applying *all possible factors at once*, and provide an *immediate solution*.

- A conventional computer uses binary bits, each of which can be in one of two states: true or false, 1 or 0. A sixteen-bit computer number can be in *one* of 2^{16} states at a particular time.
- A quantum computer uses "qubits," sort of like bits except they can be in a superposition of true and false, 1 and 0, at the *same time*. A sixteen-qubit quantum computer number is capable of being in a superposition of *2^{16} states at once*.

Quantum computers are in a very early stage of development and some serious practical issues have yet to be addressed, but the possibility exists that they will provide solutions to difficult problems like integer factorization. If this happens, we will live in a world significantly different than the present one.

How Many Primes?

Infinity

It is known that there are an infinity of primes, and this has been known since antiquity. In Euclid's Elements , Book IX, proposition 20, Euclid offers a proof similar to this one:

1. Assume there is a largest prime number. Call this number **p**.
2. Create new number **q**, equal to the *product* of all primes between 2 and **p**, plus 1.
3. Our new number **q** has no factors in the original set of primes (between 2 and **p**), because dividing by any of them would produce a remainder of 1.
4. From point (3) we conclude that **q** is either itself prime, or is composed of prime factors, all of which are larger than **p**.
5. Point (4) falsifies point (1).

Even though a proof like the above needs no specific numerical evidence, here is an example with real numbers:

1. Assume the largest prime is **17**: **p = 17**.
2. Our new number **q** equals the product of all primes in the set **{2 ... p}** plus **1**: **q = 2 * 3 * 5 * 7 * 11 * 13 * 17 + 1 = 510511**
3. According to the factoring calculator at the top of this page, **q** has these prime factors: **19, 97, 277**.
4. All the prime factors of **q** are larger than **p**.

5. Point (4) falsifies point (1).

There are a large number of proofs of the infinity of primes, this one happens to be easy to understand.

Prime Number Theorem

While running the Sieve of Eratosthenes example above, the reader may have noticed that, as ordinary numbers get larger, there are fewer prime numbers scattered among them. This observation has been studied a great deal, and has led to the following theorem:

- Let **π(x)** be a prime-counting function that gives the number of prime numbers at or below a particular real number **x**.
- Let **ln(x)** denote the natural logarithm of **x**.
- The Prime Number Theorem asserts this limit:

$$(2)\quad \lim_{x\to\infty} \frac{\pi(x)}{x/\ln(x)}=1$$

- The limit in equation (2), known as the *asymptotic law of distribution of prime numbers*, can be restated as an approximation:

$$(3)\quad \pi(x)\approx\frac{x}{\ln(x)}$$

- In everyday language, the Prime Number Theorem says that, as **x** approaches infinity, $\frac{x}{\ln(x)}$ becomes a better and better estimate of the number of primes at or below **x**.

The Prime Number Theorem is one component in an effort to understand primes and their distribution. The fact is prime numbers are still rather poorly understood, with many important questions yet to be answered. Because of our reliance on primes as shown by the widespread acceptance of RSA cryptography, we can't afford to ignore the inner workings of primes.

Conclusion

Number theory is one of many specialized fields of mathematics that until recently had no obvious practical purpose. As we come to rely more and more on technology, and as our study of nature becomes more sophisticated, more dependent on evidence and less dependent on belief, we are becoming aware that mathematics is an articulate way for us to describe nature, and for nature to describe herself to us.

Consider what is revealed by the locust example presented in this article — it means that natural processes, and natural selection, can generate prime numbers as easily as the Sieve of Eratosthenes (and by way of a similar mechanism). Until now, scientists have sometimes been caught off guard by the degree to which nature is described by mathematics, but I think we're coming to realize that mathematics is nature's native tongue, and we all should learn the language nature speaks in.

If we plan to visit France, it can't hurt to learn a few simple tourist phrases, just to get along, *n'est-ce pas?* The more French we learn, the more we will understand while traveling in France. *The same can be said of nature.* In my view, to try to understand nature without learning her native tongue seems rude.